AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method comprising:

receiving a plurality of non-native instructions in a selected one of a source form and an intermediate form; and

compiling the plurality of non-native instructions to generate object code for the non-native instructions, wherein compiling the plurality of non-native instructions includes replacing an object code segment from the generated object code with an alternative object code segment if the alternative object code segment improves at least a selected one of a power level required and an [[the]] amount of energy required to execute the generated object code in a target execution environment.

2. (Original) The method of claim 1, wherein said receiving comprises receiving the non-native instructions in a byte code form.

3. (Previously presented) The method of claim 1, wherein said compiling comprises analyzing the object code segment for execution power level requirement, and determining whether an alternative object code segment with lower execution power level requirement is available.

4. (Previously presented) The method of claim 1, wherein said compiling comprises analyzing the object code segment for execution energy consumption, and determining whether an alternative object code segment with lower execution energy consumption is available.

5. (Previously presented) The method of claim 1, wherein the method further comprises executing the non-native instructions for an initial number of times using an interpreter; and

performing said compiling only after executing the non-native instructions for said initial number of times.

6. (Previously presented) The method of claim 5, wherein the method further comprises determining the initial number of times the received non-native instructions are to be executed using the interpreter before performing compiling the received non-native instructions.

7. (Previously presented) The method of claim 6, wherein the method further comprises

monitoring said compiling for power level required to perform compilation;

updating a current understanding of power level required for compilation; and

determining the initial number of times received non-native instructions are to be executed using the interpreter before compiling the received non-native instructions, if said monitoring observes a power level required for compilation to be different from the current understanding.

8. (Previously presented) The method of claim 6, wherein the method further comprises

monitoring said compiling for amount of energy required to perform an average compilation;

updating a current understanding of amount of energy required for an average compilation; and

determining the initial number of times received non-native instructions are to be executed using the interpreter before compiling the received non-native instructions, if said monitoring observes an amount of energy required for compilation to be different from the current understanding.

9. (Original) The method of claim 1, wherein the generated object code comprises a plurality of native instructions, and the method further comprises

monitoring execution of the generated object code for power level required to execute the native instructions; and

updating power level requirements of selected ones of the native instructions if said monitoring observes power level requirements for the selected ones of the native instructions to

be different from current understandings of the power level requirements of the selected ones of the native instructions.

10. (Original) The method of claim 1, wherein the generated object code comprises a plurality of native instructions, and the method further comprises

monitoring execution of the generated object code for amount of energy required to execute the native instructions; and

updating energy requirements of selected ones of the native instructions if said monitoring observes energy requirements for the selected ones of the native instructions to be different from current understandings of the energy requirements of the selected ones of the native instructions.

11. (Currently amended) In an electronic device, a method of operation, comprising:

receiving a plurality of non-native instructions;

determining an initial number of times to interpretively execute the non-native instructions based at least in part on one or more of an expected power level required to perform a compile or an expected energy required to perform [[a]] the compile;

executing the non-native instructions for the initial number of times using an interpreter; and

compiling the non-native instructions into object code after executing the received non-native instructions for said initial number of times using the interpreter.

12. (Canceled)

13. (Previously presented) The method of claim 11, wherein the method further comprises monitoring said compiling for a compilation requirement employed in determining the initial number of times the received non-native instructions are to be executed using the interpreter before compiling; and

updating a current understanding of the compilation requirement if said monitoring observes the compilation requirement to be different from the current understanding.

14. (Original) The method of claim 11, wherein the generated object code comprises a plurality of native instructions, and the method further comprises monitoring execution of the generated object code for execution requirements of the native instructions; and

updating execution requirements of selected ones of the native instructions if said monitoring observes execution requirements for the selected ones of the native instructions to be different from current understandings of the execution requirements of the selected ones of the native instructions.

15. (Previously presented) An article of manufacture comprising:

a computer readable medium; and

a plurality of instructions designed to implement a compiler to compile non-native instructions to generate object code for the non-native instructions, and replace an object code segment with an alternative object code segment if the alternative object code segment improves at least a selected one of a power level required and an energy required to execute the generated object code.

16. (Previously presented) The article of claim 15, wherein said compiler analyzes the object code segment for execution power level requirement, and determines whether an alternative object code segment with lower execution power level requirement is available.

17. (Previously presented) The article of claim 15, wherein said compiler analyzes the object code segment for execution energy consumption, and determines whether an alternative object code segment with lower execution energy consumption is available.

18. (Currently amended) An article of manufacture comprising:

a computer readable medium; and

a plurality of instructions designed to implement a runtime manager equipped to receive a plurality of non-native instructions, determine an initial number of times to interpretively execute the non-native instructions based at least in part on one or more of an expected power level required to perform an average compile or an expected energy required to perform [[an]] the average compile, execute the non-native instructions for an said initial number of times using an interpreter, and invoke a compiler to compile the non-native instructions into object code after executing the received non-native instructions for said initial number of times using the interpreter.

19. (Canceled)

20. (Original) The article of claim 18, wherein the runtime manager is further equipped to

monitor said compiling for a compilation requirement employed in determining the initial number of times received non-native instructions are to be executed before compiling; and

update a current understanding of the compilation requirement if said monitoring observes the compilation requirement to be different from the current understanding.

21. (Original) The article of claim 18, wherein the generated object code comprises a plurality of native instructions, and the runtime manager is further equipped to monitor execution of the generated object code for execution requirements of the native instructions; and

update execution requirements of selected ones of the native instructions if said monitoring observes execution requirements for the selected ones of the native instructions to be different from current understandings of the execution requirements of the selected ones of the native instructions.

22. (Previously presented) A system, comprising:

a storage medium having stored therein a plurality of instructions implementing a compiler to compile non-native instructions to generate object code for the non-native instructions, and replace an object code segment with an alternative object code segment if the alternative object code segment improves at least a selected one of a power level required and an energy required to execute the generated object code; and

a processor coupled to the storage medium to execute the instructions implementing the compiler.

23. (Previously presented) The system of claim 22, wherein said compiler analyzes the object code segment for execution power level requirement, and determining whether an alternative object code segment with lower execution power level requirement is available.

24. (Previously presented) The system of claim 22, wherein said compiler analyzes the object code segment for execution energy consumption, and determining whether an alternative object code segment with lower execution energy consumption is available.

25. (Original) The system of claim 22, wherein the apparatus further comprises a wireless communication interface to receive the non-native instructions.

26. (Currently amended) A system, comprising:

a communication interface to receive a plurality of non-native instructions;

a storage medium coupled to the communication interface, and having stored therein a plurality of instructions designed to implement a runtime manager equipped to determine an initial number of times to interpretively execute the non-native instructions based at least in part on one or more of an expected power level required to perform an average compile or an expected energy required to perform [[an]] the average compile, execute the received non-native instructions for the initial number of times using an interpreter, and invoke a compiler to compile

the non-native instructions into object code after executing the received non-native instructions for said initial number of times using the interpreter; and

a processor coupled to the storage medium to execute the instructions implementing the runtime manager.

27. (Canceled)

28. (Previously presented) The system of claim 26, wherein the runtime manager is further equipped to monitor said compiling for a compilation requirement employed in determining the initial number of times received non-native instructions are to be executed using the interpreter before compiling; and

update a current understanding of the compilation requirement if said monitoring observes the compilation requirement to be different from the current understanding.

29. (Original) The system of claim 26, wherein the generated object code comprises a plurality of native instructions, and the runtime manager is further equipped to

monitor execution of the generated object code for execution requirements of the native instructions; and

update execution requirements of selected ones of the native instructions if said monitoring observes execution requirements for the selected ones of the native instructions to be different from current understandings of the execution requirements of the selected ones of the native instructions.

30. (Original) The system of claim 26, wherein the communication interface is a wireless communication interface.

31. (Previously presented) The method of claim 11, wherein the method further comprises determining an initial number of times to interpretively execute the non-native instructions based at least in part on a size of the non-native instructions.

32. (Previously presented) The article of claim 18, wherein the runtime manager is further equipped to determine an initial number of times to interpretively execute the non-native instructions based at least in part on a size of the non-native instructions.

33. (Previously presented) The system of claim 26, wherein the runtime manager is further equipped to determine an initial number of times to interpretively execute the non-native instructions based at least in part on a size of the non-native instructions.